"But the Auditor Said We Need to ..." Striking a Balance Between Controls and Productivity

Greg Deller
Capgemini Government Solutions

This article discusses the common gap between audit-recommended controls and those typically implemented by software project teams. The article lists commonly misunderstood audit recommendations and provides a clear explanation of what the auditors are really seeking from software project teams.

All too frequently audit recommendations are received by software project teams and immediately hung up on a dartboard for target practice. One reason they are not well received is because they are misunderstood. Audit recommendations are simply designed to be guidelines to achieve *reasonable* control, not specific instructions that hinder an organization's ability to be productive and efficient.

With the appropriate level of communication between auditors and project teams, audit recommendations can actually be extremely helpful for organizations to reduce the risk of excessive defects, delayed releases, cost overruns, and unmet customer requirements.

Unfortunately, poor communication between auditors and project teams is one of the main reasons that organizations are not able to effectively reduce their systems-based risk exposure. Frequently this poor communication causes one of two scenarios. The first is that project teams overstate the effort necessary to become compliant with audit recommendations, thinking that what the auditors are asking is entirely unfeasible. In the second scenario, project teams underestimate what the auditors are recommending because they missed the intent of the recommendation.

This miscommunication between auditors and software project teams can be thought of as *recommendation gap*. This article lists the most frequent scenarios of recommendation gap in an effort to shrink the gap. By analyzing the *intent* of the audit recommendations as opposed to their specific wording, much insight can be gained to reduce the risk of a failed project. Although this article is directed toward software projects, the viewpoints can apply to a variety of system initiatives.

Classic scenarios of overestimating the effort necessary to comply with recommendations are presented below; following these are scenarios of underestimating the effort necessary to comply with recommendations.

Classic Scenarios of Overestimating Effort

Project Plan

A very common audit finding is the lack of a project plan. Auditors recommend developing a project plan (as opposed to just a project schedule) for new software projects that are important to an organization (i.e., financially, politically, or strategically). The purpose of the project plan is to communicate to others how the project activities will be controlled.

Many project managers perceive project plans as oversized documents that no one has time to develop or read. On the contrary, they cannot afford to go without a project plan; the plan can communicate necessary information to the project team instead of project managers communicating the information multiple times or not communicating the information at all, thus creating chaos. It should also be noted that an indirect benefit to creating and maintaining a project plan is that it forces project management to think about critical topics otherwise not considered.

Auditors are not looking for the project plan to meet a certain length or depth requirement; they are just concerned that pertinent components are addressed (see Table 1 for potential components). For instance, depending on the size of a project it is conceivable that topics such as stakeholder identification and interaction can be addressed in a paragraph.

Table 1: *Recommended Project Plan Components*

Recommended Project Plan Components	
System Life-Cycle Considerations	Milestones
Technical and Management Tasks	Data Management
Budgets and Schedules	Risk Identification
Resource and Skill Requirements	Stakeholder Identification and Interaction

System Life-Cycle Methodology

Another very common audit recommendation is for organizations to adopt and implement a formal system life-cycle methodology. Organizations tend to take the extreme of this recommendation by either adopting a robust yet unwieldy system life-cycle methodology that virtually no one is trained on, or ignoring the recommendation altogether, seeing it as a mountainous task.

This recommendation can be addressed with relative ease by a project team considering its strengths and weaknesses and adopting a basic life-cycle methodology that exploits their strengths and overcomes their weaknesses. There are a number of proven life-cycle methodologies that are publicly available; creating one from scratch is generally unnecessary. Widely used methodologies include waterfall, rapid prototyping, incremental build, multiple build, spiral, fast-track, and hybrid.

Project teams should select a methodology that can be easily implemented and will not require extensive training. This leads to an area that most organizations fall short on – training and implementing the methodology. Auditors are not concerned that a methodology that fills five binders has been developed. Instead, they want to see that a valid methodology appropriate for the specific project team has been adopted and implemented.

Change Management

Change management policies and procedures do not need to handcuff an organization's systems' staff, but they should be commensurate with the risks of the specific environment. Modifications to a missile control system present a higher risk than modifications to a video library system; therefore, change management policies and procedures must be comparatively more structured. Auditors simply want to see that a change management process is carefully designed for the specific operational and technology environment as well as being implemented entity-wide.

An important component of this

includes the organization's ability to ensure "consistency in the management and control of software changes" [1]. This begins with establishing a policy and procedure describing how program changes are to be made. These policies and procedures should incorporate a structured process to ensure that system changes are requested, tested, and approved prior to implementation, as opposed to an informal approach (such as ad-hoc troubleshooting).

While a structured process does add new requirements, it can be implemented without requiring a significant increase in resources, if designed appropriately.

> Increasingly, organizations are recognizing that effective change control management is a key to assuring that the product delivered is indeed the product intended and expected. [1]

If the product is delivered as intended and expected the first time, fewer resources are needed in the long run to fix the problem.

Disaster Recovery

Disaster recovery considerations should be addressed from the inception of a project. Should the Internal Revenue Service cut over to an Internet-based tax filing system without considering high availability? Should the Navy implement a new fleet maintenance tracking system without considering a back-up scheme? The answers are obvious.

On the other end of the spectrum, does a small construction company need to consider disaster recovery for their Internet access? They do if they are implementing a new payroll system utilizing a third-party payroll processor that can only receive payroll submissions via a file transfer from a specific Internet Protocol address.

All of these scenarios present a risk of continued operations that should be discussed among the project team and stakeholders prior to the implementation date. However, if the topic is not discussed during the early stages of the project, implementation decisions may be made about the system that negatively affect disaster recovery capabilities. Disaster scenarios and contingencies should at least be discussed and documented among all key personnel when developing a new system. This can be as simple as brainstorming about the risks, mitigating circumstances, and contingencies during a team meeting at key milestones during the project.

Audit requirements should not dictate across the board that systems are fully redundant and disaster recovery plans meet the five-binder requirement. What they do require is that disaster risks are identified and the impact of the risks is assessed. Based on the risk analysis and business impact analysis, management can then determine what level of risk they are willing to accept versus the cost. Reasonable contingencies should be identified and implemented to reduce the overall risk exposure. Again, these planning exercises may be as simple as getting the appropriate personnel in a room to discuss and document the decisions.

Further, there is a common misconception within organizations regarding who should be responsible for addressing business continuity and disaster recovery. The key is not who is responsible for it or who manages the effort, but who provides input to the disaster recovery planning effort. Input to the disaster recovery effort must come from all stakeholders including management, end users, and systems personnel. It should be a coordinated effort where business and technical issues are discussed.

"Unfortunately, poor communication between auditors and project teams is one of the main reasons that organizations are not able to effectively reduce their systems-based risk exposure."

Data Conversion

Auditors typically recommend a structured approach be taken with most complex project tasks. Data conversion is no exception. The *recommendation gap* in this area is generally more prevalent for small- and medium-sized projects than with large projects. Large projects generally identify a data conversion approach and perform data conversion activities that support the approach. These data conversion activities can include development of the following documentation:

- List of all of the legacy data that must be cleansed and loaded into the new system.
- Mapping diagrams for data from the legacy system to the new system.
- Conversion plan, which includes the approach (e.g., manual entry, file load, transaction load, automated program, etc.).
- Data conversion reconciliation and balancing procedures.
- Error resolution for data conversion errors.
- Post-migration review and approval from an appropriate stakeholder.

Although these items take a significant amount of effort to prepare for a large project, they can be prepared for small- and medium-sized projects with limited resources by focusing on the intent of the documents. The intent is to develop an effective and controlled approach to data conversion. An important element of a controlled approach is that it be well formulated and communicated to all involved, thus the need for documentation (e.g., plans, mapping diagrams, error resolution, etc.).

Again, there is no requirement regarding the length or depth of the data conversion documentation, only that it addresses the key decisions (e.g., conversion method, categories of data converted) and documents the reconciliation process so it can be re-performed.

End-User Testing

End-user testing can go a long way toward developing a relationship with customers and gaining their support. But primarily end-user testing is designed to catch defects by using a tester that may be more familiar with the subject matter and provide a fresh set of eyes.

> User acceptance testing identifies defects before they get into production and gives the user community a chance to *kick the tires* on the system before it goes live. [2]

Why wait until a system is already implemented to learn that the menu names do not meet user needs, or that a system formula for a calculation is incorrect?

End-user testing can be as simple as having a sample of end users identify their key activities and execute them in the test system, or by having end users review system-generated reports for validation of accuracy and effectiveness. The key to end-user testing is user confirmation of the accuracy and functionality of the system. This can be easily accomplished by working with the users to list the key activities that they perform (i.e., test items) and

	Security Role (User Profile)			
Functions	Clerk	Accountant	Supervisor	Controller
Journal Entry (JE1)	✓	✓	✓	
Journal Inquiry (JI1)	✓	✓	✓	✓
Journal Post (JP1)		✓	✓	
Journal Reporting (JR1)	✓	✓	✓	✓

Note: The screen name is listed in parentheses.

Table 2: *Security Matrix*

request that they test the key activities in the system for accuracy and effectiveness.

During the end users' test process they should document their validation of each of the key processes listed and clearly document any errors or problems. Not only does the list help the users know what to test, but it indirectly helps to ensure that they perform each of the components of the test. Auditors' primary concern is that end users perform key process testing and confirm the accuracy and functionality prior to implementing a system, not that the end users test every intricate element of a system.

Go-Live Approval

End-user management and project team management must perform an exercise to identify criteria that must be satisfied prior to implementation of a change into the live environment. They must then review compliance with the criteria and collectively approve a system's readiness prior to implementation. However, this may be as simple as gathering the appropriate people in a meeting to gain their documented approval that go-live criteria have been satisfied.

Auditors are not as concerned about the details of the go-live review as they are about the process of evaluation. This includes how the go-live criteria were identified and the process to review and approve compliance with the go-live criteria. Accountability and ownership for the go-live decision can be provided by end-user management and project team management signing the go-live criteria checklist and maintaining it with the central project files.

Classic Scenarios of Underestimating Effort

The discussion will now switch to scenarios where project teams frequently underestimate the level of effort required to meet audit recommendations. Again, the cause of this recommendation gap is attributed to miscommunication regarding the *intent* of the auditors' recommendations.

Application Security

Project teams have a tendency to be focused on ensuring that the requested system functionality is implemented within, generally, a tight timeframe. Often, that leaves application security as merely an afterthought or a task that is quickly addressed by a small team, not consisting of appropriate personnel, based primarily on assumptions of needed user access.

Instead of addressing security at the tail-end of the project, auditors recommend that security be addressed at each phase of the project, including project initiation and planning. From a logical perspective, this entails appropriate personnel identifying sensitive data and function access, establishing roles (i.e., user profiles) for users, and then assigning data and function access to the roles. From a physical perspective, application security design must be performed simultaneously and integrated with the system design process to help eliminate downstream conflicts.

An accounting-based example of this conflict exists when a user needs access to *Journal Inquiry* on a screen, but should be prevented from accessing *Journal Entry* on the same screen (assuming security is restricted at the screen level). A possible reason for this conflict is that system designers did not consider security considerations when they were designing the system so it did not occur to them that the two incompatible functions should be on different screens.

Logical security specifications can be documented in a security matrix (for example, refer to Table 2). In a security matrix, groupings of users that have the same security requirements are formed into security roles. Then specific functions in the application are assigned to the security roles. Depending on the implementation, the security matrix can even be used to document the mapping from the logical design to the physical design if the function column is granular enough to match physical application security (e.g., screen names).

Independent Migration

Although audit recommendations repeatedly suggest that all code changes be migrated to the live (production) environment by an independent person, that alone is not sufficient. The intent of the independent migration is so that an independent person can review the change and ensure that it has been through proper testing, documentation, and review prior to implementation. However, limited benefit is gained by having an independent person migrate the change to the live environment if he or she does not perform a review or provide oversight for the change. This independent review applies to any program changes to the live environment, regardless of the source of the change (e.g., development/test environment).

Varying levels of review can be performed by this independent person, but the goal is for the independent person to ensure the appropriateness of the change. As an example, the independent reviewer could verify that a Change Control Board has authorized the change, or the independent reviewer could be responsible for performing his or her own review and test of the code. Then the independent reviewer is responsible for raising the issue through proper channels if the change is not appropriate.

Risk Management Plan

Although risk management planning is gaining attention with large projects, it is still rarely addressed in small- or medium-size projects. However, even small projects developed in the current economy have management visibility that justifies the need for risk management.

> The first objective in a risk management plan is to prevent undesirable situations from occurring. The second objective is to reduce any negative consequences when something undesirable does occur. [3]

For example, an organization may address the first objective by aggressively compensating employees who are the sole provider of a skill to prevent their departure. Then, they may address the second objective by cross-training employees to reduce the negative consequence if the resource does leave the organization. Note that disaster recovery planning meets the second objective of risk management because it aims to reduce any negative consequences when something undesirable does occur.

As Dr. Richard Bechtold indicates, a plan should be developed that addresses how the project team does the following:

> … intends to identify, evaluate, prioritize, mitigate, and manage project risks. Select the top five or 10 risks to be the primary focus of your risk management activities

and describe each. Document the probability and impact of each risk and calculate the resulting risk exposure. [3]

This traditionally has been a topic that is addressed informally by project managers that believe they can see problems coming on the horizon and perceive that their projects are small enough to have visibility to all of the risks. Instead, a proactive approach to risk management is typically recommended by auditors even for small- and medium-size projects. The designation of resources to perform risk management activities must be performed during project planning to avoid the scenario of risk management activities becoming a drain on the project team's resources.

Closing the Recommendation Gap

This article has mentioned the recommendation gap several times to convey the prevalence of poor communication between auditors and project teams regarding audit recommendations. While there is no doubt that both sides are responsible for fostering open and honest communication, the auditors are primarily responsible for ensuring that recommendations are clearly communicated and there is no confusion about the intent of the recommendations. The following list provides actions that can be performed to close the recommendation gap:

- Project teams can request to have daily or weekly briefings on issues or audit concerns that arise during the audit. Also, the two parties should meet at the end of an audit to verbally discuss all recommendations prior to a report being finalized.
- Auditors can provide the project teams with examples and templates of documents that they recommend the project teams develop.
- Auditors can brief line management on the details of the recommendations prior to briefing upper management since the line managers will typically be directly responsible for addressing the recommendations.
- Project teams can request clarification (verbal or written) on audit reports prior to devising responses or action items.
- Auditors can provide the source for their recommendation such as the Software Engineering Institute's Capability Maturity Model®, Project Management Institute standards, Control Objectives for Information and Related Technology Audit Guidelines, Federal Information Processing Standards publications, American Institute of Certified Public Accountants standards, and Financial Accounting Standards Board (or other regulators) standards.

Summary

There is a common gap between audit-recommended controls and those typically implemented by software project teams. Often, that recommendation gap can be attributed to miscommunication between the auditors and those being audited. Miscommunication results from the project team either overestimating or underestimating the effort needed to comply with audit recommendations. Extensive communication between the auditors and project teams is necessary to close the recommendation gap. Experience shows that the implementation of audit recommendations will reduce risk and can also lead to improved efficiency and effectiveness.◆

References

1. Vallabhaneni, S. Rao. Certified Information Systems Auditor Examination Textbook Vol. 1: Theory. 2nd ed. Los Angeles, CA: SRV Professional Publications, 1996.
2. Mogyorodi, Gary E. "Let's Play 20 Questions: Tell Me About Your Organization's Quality Assurance and Testing." CROSSTALK. Mar 2003: 30.
3. Bechtold Ph.D., Richard. Essentials of Software Project Management. Vienna, VA: Management Concepts, 1999: 110.

About the Author

Greg Deller is employed by Capgemini Government Solutions. Previously, he was a manager with KPMG's Information Risk Management practice. Deller is a Certified Information Systems Auditor with more than eight years of systems-based risk management and consulting experience. He has advised over 110 clients on the security and controls in their information systems environment. Deller has a Master of Science in information systems from George Mason University in Fairfax, Va.

Phone: (703) 244-5202
Fax: (208) 723-1537
E-mail: dellerg@yahoo.com